République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

**UNIVERSITE M'HAMED BOUGARA – BOUMERDES**

A logo with text and a sun

AI-generated content may be incorrect.

Faculté des Sciences

Département d’Informatique

**Domaine :** Mathématiques Informatique

**Filière :** Informatique

**Spécialité :** Ingénierie des systèmes d’information et du logiciel

***Mémoire de fin d’études***

En vu de l’obtention du Diplôme de Licence Académique

***Thème***

**Développement d'une plateforme web pour le département informatique**

**Encadré par :**

Mme: ALOUANE Besma

**Réalisé par :**

GOUFFI Mohamed Ryad

SALHI Nabil

REFADA Imad Eddine

***Remerciement***

Avant tout, nous rendons grâce à **Allah Tout-Puissant**, qui nous a accordé la force et la patience nécessaires pour mener à bien ce projet.

Nous adressons nos sincères remerciements à **Mme ALOUANE Basma**, notre encadrante, pour son soutien, sa disponibilité et ses précieux conseils tout au long de la réalisation de ce travail. Son accompagnement nous a été d’une grande aide et nous a permis d’avancer avec confiance.

Nous exprimons également notre profonde gratitude à nos **parents et proches**, pour leur soutien inconditionnel, leur encouragement et leur patience durant cette période exigeante.

Enfin, nous souhaitons remercier chaleureusement toutes les personnes qui, de près ou de loin, ont contribué à la réussite de ce projet.

***Ryad et Nabil et Imad***

***Dédicaces***

À mes parents, pour leur amour et leurs sacrifices. À mes sœurs, pour leur soutien et leur présence. Et à mes amis, pour leur encouragement et leur aide.

**GOUFFI Mohamed Ryad**

À ma famille, pour leur confiance et leur patience et leur couscous, et à mes amis, pour les moments de joie et de motivation partagés.

**SALHI NABIL**

À mes parents, mes frères et sœurs, pour leur soutien inconditionnel, et à mes amis, pour leur présence et leur aide précieuse.

**REFADA Imad Eddine**

**Table des Matières**

[**Introduction Générale :** 1](#_Toc198228357)

[Chapitre 1 : Contexte du travail 2](#_Toc198228358)

[1.1 Introduction 2](#_Toc198228359)

[1.2 Présentation du Département d’Informatique 2](#_Toc198228360)

[1.3 Problématique 3](#_Toc198228361)

[1.3.1 Problèmes Identifiés : 3](#_Toc198228362)

[1.3.2 Conséquences : 4](#_Toc198228363)

[1.4 Solution Proposée 4](#_Toc198228364)

[1.5 Objectifs du Projet 4](#_Toc198228365)

[1.6 Conclusion 5](#_Toc198228366)

[Chapitre 2 : Conception et Modélisation 6](#_Toc198228367)

[2.1 Introduction 6](#_Toc198228368)

[2.2 Présentation de la méthode UML 6](#_Toc198228369)

[2.2.1 Structure d’UML 6](#_Toc198228370)

[2.3 Analyse des bosoins 8](#_Toc198228371)

[2.3.1 Besoins non Fonctionnels 8](#_Toc198228372)

[2.3.2 Besoins Fonctionnels 8](#_Toc198228373)

[2.4 Etude des acteurs 8](#_Toc198228374)

[2.5 Diagramme de cas d’utilisation 9](#_Toc198228375)

[2.5.1 Diagramme de cas d’utilisation général 10](#_Toc198228376)

[2.5.2 Diagramme de cas d’utilisation détaillé 11](#_Toc198228377)

[2.6 Diagramme de séquence 14](#_Toc198228378)

[2.6.1 Definition 14](#_Toc198228379)

[2.6.2 Diagramme de séquence de cas d’utilisation <<authentification>> 16](#_Toc198228380)

[2.6.3 Diagramme de séquence de cas d’utilisation <<Déposé les notes>> 17](#_Toc198228381)

[2.6.4 Diagramme de séquence de cas d’utilisation <<Demande de document>> 17](#_Toc198228382)

[2.6.5 Diagramme de séquence de cas d’utilisation <<Gérer demande de document>> 18](#_Toc198228383)

[2.6.6 Diagramme de séquence de cas d’utilisation <<Déposer cour>> 18](#_Toc198228384)

[2.7 dictionnaire des données 19](#_Toc198228385)

[2.8 Règles de transformation du diagramme de classes en modèle relationnel 21](#_Toc198228386)

[2.9 Diagramme de classe 23](#_Toc198228387)

[2.10 Modèle relationnel 24](#_Toc198228388)

[2.11 Conclusion 24](#_Toc198228389)

[Chapitre 3 : Implémentation 25](#_Toc198228390)

[3.1 Intoduction 25](#_Toc198228391)

[3.2 Outils, langages et technologies de développement 25](#_Toc198228392)

[3.2.1 Les langages de balisage, de style et de programmation 25](#_Toc198228393)

[3.2.2 Environnements de développement et outils utilisés 26](#_Toc198228394)

[3.3 Interfaces de l’application 27](#_Toc198228395)

**Tables des Figures**

[Figure ‎1.2‑2 Organigramme de la Faculté des Sciences 2](#_Toc198228402)

[Figure ‎2.2‑1logo UML 6](#_Toc198228403)

[Figure ‎2.5‑1 les elements de tableau de cas d’utilisation 9](#_Toc198228404)

[Figure ‎2.5‑2 Diagramme de cas d’utilisation général 10](#_Toc198228405)

[Figure ‎2.5‑3 cas d'utilisation <<admin>> 11](#_Toc198228406)

[Figure ‎2.5‑4 cas d'utilisation <<enseignant>> 12](#_Toc198228407)

[Figure ‎2.5‑5 cas d'utilisation <<student>> 13](#_Toc198228408)

[Figure ‎2.6‑1les principaux concepts du diagramme de séquence 15](#_Toc198228409)

[Figure ‎2.6‑2 Diagramme de séquence de cas d’utilisation <<authentification>> 16](#_Toc198228410)

[Figure ‎2.6‑3 Diagramme de séquence de cas d’utilisation <<Déposé les notes>> 17](#_Toc198228411)

[Figure ‎2.6‑4 Diagramme de séquence de cas d’utilisation <<Demande de document>> 17](#_Toc198228412)

[Figure ‎2.6‑5 Diagramme de séquence de cas d’utilisation <<Gérer demande de document>> 18](#_Toc198228413)

[Figure ‎2.6‑6 Diagramme de séquence de cas d’utilisation <<Déposer cour>> 18](#_Toc198228414)

[Figure ‎2.10‑1 Description des tables de la BDD (MYSQL) 21](#_Toc198228415)

[Figure ‎2.7‑1 Diagramme de classe 23](#_Toc198228416)

**Introduction Générale :**

Notre projet consiste à développer une plateforme web pour le département d’informatique de la faculté des sciences de l’université UMBB. Cette plateforme vise à simplifier la gestion des informations, améliorer la communication entre les enseignants et les étudiants, et offrir un espace centralisé pour les ressources pédagogiques.

Ce projet s’inscrit dans une démarche de modernisation et d’innovation, afin de rendre les processus du département plus efficaces et adaptés aux besoins actuels. Il représente une étape importante dans la digitalisation des services universitaires et une opportunité pour améliorer l’expérience de tous les acteurs du département.

Ce rapport est organisé en trois chapitres principaux :

* **Chapitre 1 :** présente le contexte du projet, l’organisation d’accueil (l’université UMBB et son département d’informatique), ainsi que la problématique et les objectifs de notre plateforme web.
* **Chapitre 2 :** se concentre sur la conception du système, incluant l’analyse des besoins et la modélisation à travers des diagrammes UML tels que les cas d’utilisation, les séquences et les classes.
* **Chapitre 3 :** présente la réalisation de la plateforme, incluant les technologies utilisées (langages de programmation, logiciels, serveurs) et les interfaces principales de l’application.

Enfin, une conclusion générale viendra clore ce rapport. Elle permettra de faire un bilan global du projet et de souligner les résultats obtenus.

# Contexte du travail

## Introduction

Dans ce chapitre, nous présentons le département d’informatique d’Université M’hamed Bougara de Boumerdès **(UMBB)**. Nous décrirons ensuite la **problématique** qui justifie notre projet, la **solution** proposée et les **objectifs** que nous visons à atteindre. Ce chapitre vise à fournir un contexte clair pour comprendre les enjeux et les motivations derrière la conception de notre plateforme web.

## **Présentation du Département d’Informatique**

Le département d’informatique de l’Université M’hamed Bougara de Boumerdès (UMBB) fait partie de la Faculté des Sciences. Créé en 1998, il proposait initialement des formations en DEUA et en ingénierie d’État. Avec l’introduction du système LMD en 2004, il s’est élargi pour inclure des formations en Licence, Master et Doctorat.

Le département dispense un enseignement couvrant divers domaines de l’informatique, alliant théorie et pratique, afin de préparer les étudiants aux exigences du marché du travail et de la recherche scientifique. C’est dans ce cadre que s’inscrit notre projet, visant à répondre aux besoins spécifiques de cette structure.

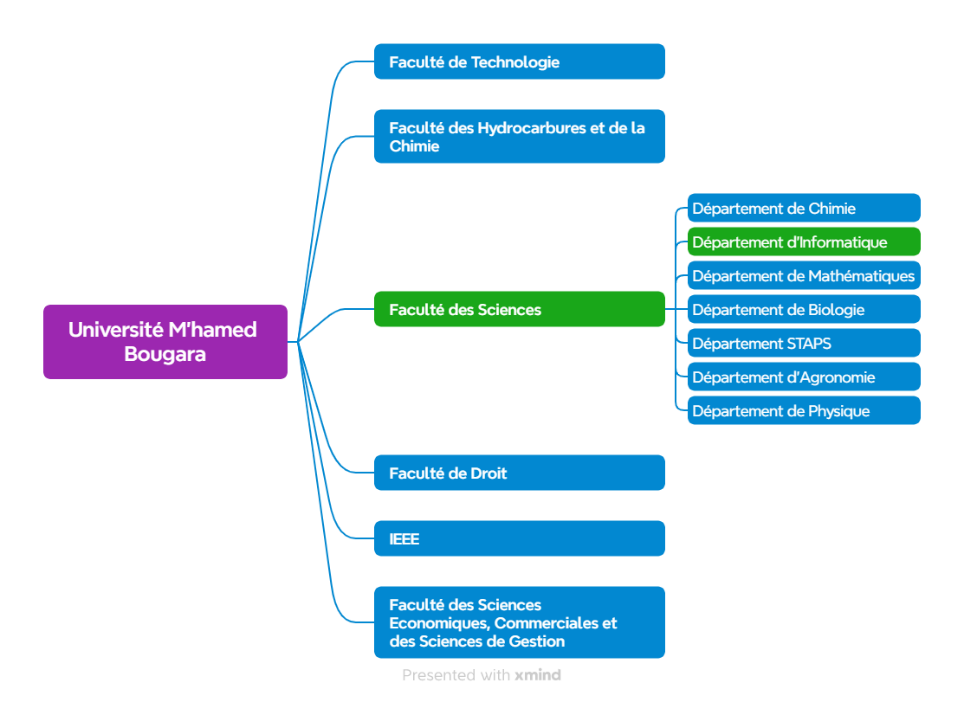




Figure ‎1.2‑2 Organigramme de la Faculté des Sciences

## Problématique

Actuellement, le département d’informatique de l’Université M’hamed Bougara de Boumerdès (UMBB) rencontre plusieurs défis liés à la communication et à la gestion des informations. Ces problèmes sont principalement dus à l’utilisation de méthodes traditionnelles et décentralisées, qui ne répondent plus aux besoins modernes d’un département dynamique et en croissance.

### Problèmes Identifiés :

1. **Communication Obsolète :**
   * Le département utilise principalement Facebook pour diffuser des annonces et des nouvelles. Cependant, de nombreux étudiants n’ont pas de compte Facebook ou ne l’utilisent pas régulièrement, ce qui entraîne un manque d’accès aux informations importantes.
   * **Absence de structuration :** Les informations partagées sous forme de publications se perdent rapidement dans le fil d’actualité, rendant la recherche d’une annonce spécifique compliquée.
   * **Dépendance à une plateforme externe :** L’utilisation d’un réseau social comme principal canal de communication académique n’est pas idéale pour une institution universitaire.
2. **Dispersion des Ressources Pédagogiques :**
   * Les supports de notes, règlements et plannings sont souvent partagés via plusieurs canaux (Facebook, Google Drive, Telegram), rendant leur accès peu pratique et désorganisé.
   * Les étudiants doivent naviguer entre différentes plateformes pour trouver les informations dont ils ont besoin, ce qui entraîne une perte de temps et de frustration.
3. **Absence de Modernisation :**
   * Les processus actuels ne sont pas adaptés aux technologies modernes, ce qui entraîne une inefficacité générale.
   * Par exemple, les annonces importantes (annulations de cours, changements d’emploi du temps) ne sont pas toujours communiquées de manière fiable ou rapide.

### Conséquences :

* **Pour les étudiants :** Difficulté à accéder aux informations en temps réel, manque d’un espace centralisé pour les ressources pédagogiques, et dépendance excessive à des plateformes non officielles (comme Facebook).
* **Pour les enseignants :** Charge de travail accrue due à la communication inefficace avec les étudiants et à la gestion manuelle des annonces.
* **Pour le département :** Image peu professionnelle due à l’utilisation de méthodes obsolètes, et difficulté à suivre l’évolution des besoins des étudiants et des enseignants.

## Solution Proposée

Pour remédier à ces problèmes, nous proposons la mise en place d’une **plateforme web dédiée au département d’informatique**. Cette plateforme centralisera l’accès aux annonces, ressources et documents importants pour les étudiants. Elle offrira une interface organisée et facile d’utilisation, permettant ainsi une meilleure accessibilité aux informations académiques essentielles.

## Objectifs du Projet

L'objectif principal de ce projet est de développer une plateforme web centralisée pour le département d’informatique de l’UMBB. Cette plateforme vise à améliorer l’accès aux informations académiques, faciliter la communication entre étudiants et enseignants, et offrir un espace structuré pour la gestion des ressources et annonces importantes.

Les objectifs spécifiques du projet sont les suivants :

* **Centralisation des informations** : Regrouper toutes les annonces, documents et ressources académiques sur une seule plateforme accessible à tous les étudiants.
* **Amélioration de la communication** : Permettre aux étudiants d’accéder facilement aux emails des enseignants et aux annonces officielles sans passer par des réseaux sociaux externes.
* **Affichage des notes et des plannings** : Offrir aux étudiants un moyen rapide de consulter leurs résultats académiques ainsi que les emplois du temps et plannings d’examens.
* **Accès structuré aux ressources pédagogiques** : Faciliter le téléchargement des cours, TD, TP et autres documents académiques via une interface organisée.
* **Annonce des événements et opportunités** : Créer un espace dédié aux compétitions, conférences, recrutements et autres événements académiques.
* **Facilitation des démarches administratives** : Intégrer un système de demande en ligne pour certains documents officiels.

En mettant en place cette plateforme, nous visons à offrir aux étudiants une solution moderne, ergonomique et efficace pour accéder aux informations essentielles de leur département.

## Conclusion

Dans ce chapitre, nous avons défini le cadre général de notre projet en présentant l’université, la faculté et plus particulièrement le département d’informatique, qui constitue notre domaine d’étude. Nous avons également mis en évidence les problèmes rencontrés dans la gestion et la communication des informations au sein du département, justifiant ainsi la nécessité de développer une plateforme web dédiée.

Afin de répondre aux besoins identifiés, nous avons établi les objectifs du projet, qui guideront la conception et le développement de la plateforme. Celle-ci vise à centraliser l’accès aux informations essentielles pour les étudiants et enseignants, en offrant des fonctionnalités adaptées aux exigences du département.

Dans le prochain chapitre, nous allons détailler la conception de notre plateforme en présentant les différents modèles et diagrammes qui illustrent son architecture et son fonctionnement.

# Conception et Modélisation

## Introduction

Dans ce chapitre, nous présenterons le processus de conception et de modélisation de notre système de gestion PFE, qui nous permet de comprendre les besoins des utilisateurs et de déterminer les exigences. Nous commencerons par présenter la méthode UML, puis nous passerons à l'implémentation de différents diagrammes, tels que les cas d'utilisation, les séquences, les activités et les classes, puis au modèle relationnel de données, ce qui nous permet d'affiner le périmètre de notre projet.

## Présentation de la méthode UML

UML (*Unified Modeling Language*, ou Langage de Modélisation Unifié)

est un langage standardisé utilisé en ingénierie logicielle pour

la modélisation et la conception de systèmes informatiques. Il permet

de représenter visuellement l'architecture, les interactions et le comportement

d'un système à l'aide de différents types de diagrammes (diagramme de classes,

Figure ‎2.2‑1logo UML

de cas d'utilisation, de séquence, d'activités, etc.). UML est largement utilisé pour faciliter la communication entre les développeurs, analystes et autres parties prenantes d'un projet logiciel.

### Structure d’UML

La structure d'UML (*Unified Modeling Language*) repose sur un ensemble de concepts et de diagrammes organisés en trois grandes catégories :

#### Diagrammes structurels (Structural Diagrams)

Ces diagrammes permettent de modéliser l'organisation statique d'un système, notamment ses composants et leurs relations. Ils incluent :

* **Diagramme de classes** : représente les classes, leurs attributs, méthodes et relations.
* **Diagramme d'objets** : montre une instance concrète des classes et leurs relations à un moment donné.
* **Diagramme de composants** : décrit les différents modules logiciels et leurs interactions.
* **Diagramme de déploiement** : représente l’architecture physique du système (serveurs, bases de données, etc.).
* **Diagramme de structure composite** : détaille l’organisation interne des classes et objets.
* **Diagramme de packages** : organise les éléments du modèle en groupes logiques.

#### Diagrammes comportementaux (Behavioral Diagrams)

Ces diagrammes décrivent le comportement du système et les interactions entre ses éléments. Ils incluent :

* **Diagramme de cas d'utilisation** : illustre les interactions entre les acteurs et le système.
* **Diagramme d’activités** : représente le flux de contrôle ou de données dans un processus.
* **Diagramme d’état (ou état-transitions)** : décrit les états possibles d’un objet et leurs transitions.

#### Diagrammes d’interaction (Interaction Diagrams)

Ils sont une sous-catégorie des diagrammes comportementaux et mettent l'accent sur les échanges entre objets :

* **Diagramme de séquence** : montre l’ordre des messages échangés entre les objets au cours du temps.
* **Diagramme de communication** : met en évidence les relations entre objets avec les messages échangés.
* **Diagramme de temps** : représente l’évolution des objets en fonction du temps.
* **Diagramme d’interaction générale** : généralise les interactions entre plusieurs éléments d’un système.

En résumé, UML est un langage de modélisation polyvalent utilisé pour décrire et communiquer la complexité des systèmes logiciels. Il propose un ensemble de diagrammes permettant de représenter la structure et le comportement d'un système, facilitant ainsi la comparaison et l'évaluation de différentes solutions. Associé à la méthode RUP, UML offre un cadre de développement logiciel garantissant la prise en compte de tous les aspects du processus et la qualité du produit final.

## Analyse des bosoins

Dans cette section, nous présentons les exigences fonctionnelles et non fonctionnelles de notre site web :

### Besoins non Fonctionnels

● Performance : Le site devra être performant, C’est-à-dire que l’application doit répondre à toutes les exigences des utilisateurs d’une manière optimale.

● Sécurité : Le besoin en sécurité peut définir les niveaux d’accès possible au système pour les utilisateurs de l’application (imposer l’authentification à chaque utilisateur).

● Maintenabilité : Le code doit être clair, c’est-à-dire qu’il doit y avoir une possibilité d’ajouter de nouvelles fonctionnalités ou de modifier celles existantes.

● Ergonomie : L’application offre une interface simple et facile à utiliser afin que l’utilisateur puisse l’exploiter sans se référer à des connaissances particulières. En d’autres termes, notre site doit être lisible et facile à manipuler par n’importe quel utilisateur.

### Besoins Fonctionnels

* Gestion des events (partage et consultation des events).
* Gestion des news (partage et consultation des news).
* Gestion des utilisateurs par l’administration.
* Gestion des notes, délibération, planning (affichage et téléchargement).
* Gestion des demandes de documents importants pour les étudiants.
* Accès aux règlements du département et emails des enseignants.

## Etude des acteurs

Un acteur représente un rôle joué par une personne externe, ou autre système qui interagit avec le système.

Pour notre cas, nous avons identifié 3 acteurs, le tableau Ci-dessous résume les acteurs et leurs tâches assurées :

|  |  |
| --- | --- |
| **Acteur** | **rôle** |
| Etudiant | - L’étudiant peut consulter les annonces, télécharger les ressources pédagogiques, voir les notes et plannings (EDT & examens), demander un document officiel, accéder aux règlements du département, accéder aux emails des enseignants. |
| Enseignant | - L'enseignant peut publier une news, partager les notes. |
| Administrateur | - L’administrateur peut gérer les comptes d’utilisateurs, gérer les events, partager les délibération et plannings (EDT & examens), valider les demandes de documents. |

## Diagramme de cas d’utilisation

Un **diagramme de cas d'utilisation** (ou *use case diagram* en anglais) est un outil de modélisation **UML** (Unified Modeling Language) qui permet de représenter les interactions entre les **utilisateurs (**ou "acteurs"**)** et un **système**. Il décrit **les fonctionnalités principales** que le système doit offrir, sans entrer dans les détails techniques de leur mise en œuvre.

|  |  |  |
| --- | --- | --- |
| **element** | **symbole** | **définition** |
| acteur |  | Entité externe (personne, système…) qui interagit avec le système. |
| cas d’utilisation |  | Fonctionnalité ou service offert par le système. |
| association | A blue and white circle with a black circle and a blue line  AI-generated content may be incorrect. | Représente une interaction ou communication. |
| inclusion |  | Un cas d’utilisation inclut systématiquement un autre (réutilisation de comportement). |
| généralisation |  | Un cas d’utilisation optionnel s'ajoute dans certaines conditions. |
| extension |  | Relation d’héritage entre acteurs ou cas d’utilisation. |

Figure ‎2.5‑1 les elements de tableau de cas d’utilisation

### Diagramme de cas d’utilisation général

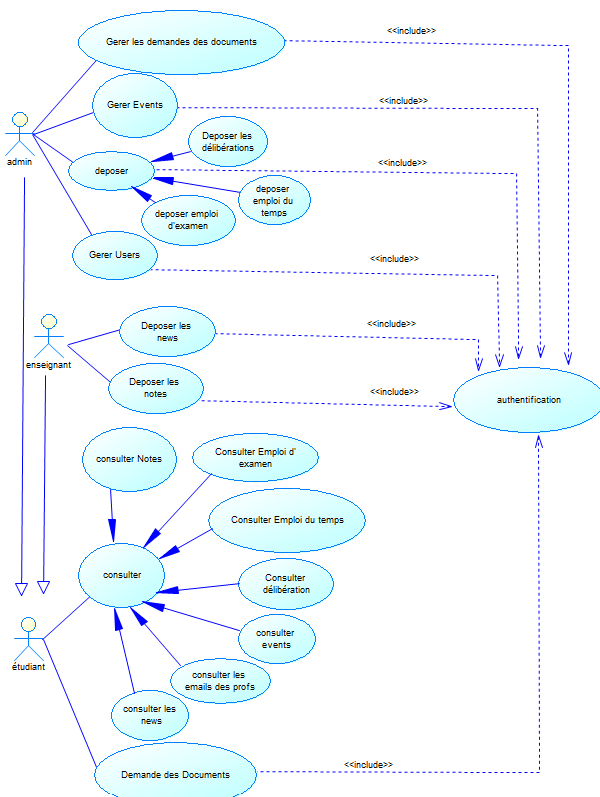


Figure ‎2.5‑2 Diagramme de cas d’utilisation général

### Diagramme de cas d’utilisation détaillé

#### Diagramme de cas d’utilisation d’admin

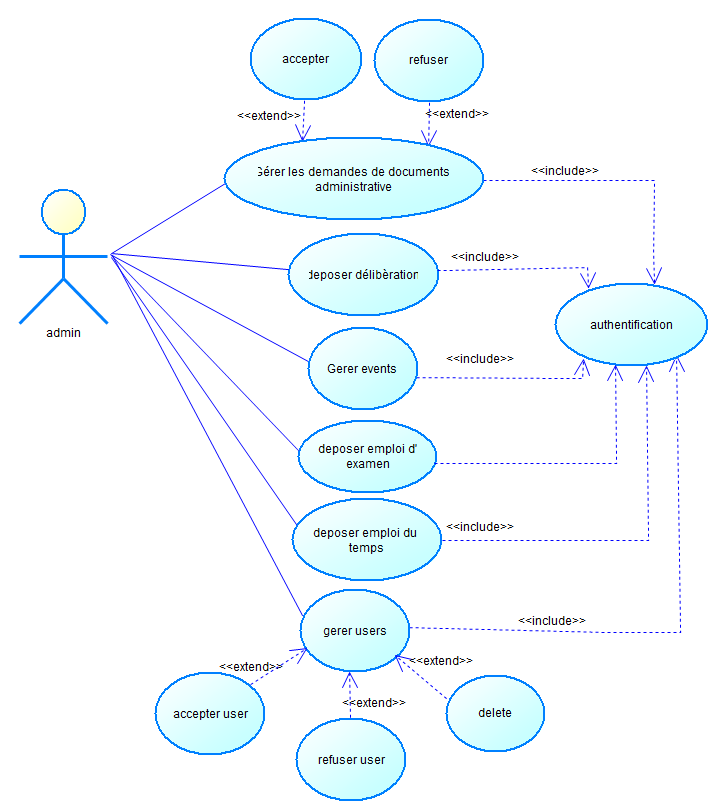


Figure ‎2.5‑3 cas d'utilisation <<admin>>

Le schéma de la figure (2.5-3) illustre le cas d'utilisation « Admin » et les fonctionnalités réalisables par l'administrateur dans le système. Le tableau suivant présente la description de ce cas d'utilisation.

|  |  |
| --- | --- |
| **Case** | **Description** |
| Authentification | L'utilisateur s'authentifie en tant qu'administrateur pour accéder au site Web |
| gestion des events | L'admin gère les événements académiques |
| deposement de délibération | L'admin partage les délibération des étudiants |
| deposement des plannings | L'admin publie les plannings(EDT,Examens) |
| traitement des documents | L'admin traite les demandes de documents |
| Gestion des comptes | L’admin accepter,refuser,supprimer des comptes (étudiants, enseignants, admin) |

#### Diagramme de cas d’utilisation d’enseignant

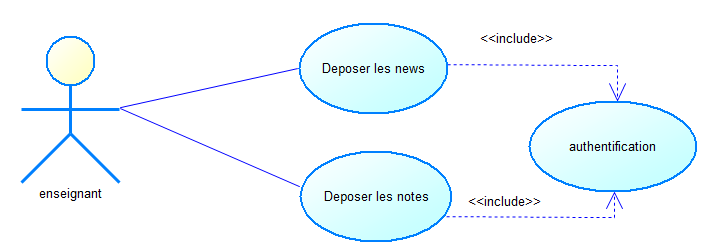


Figure ‎2.5‑4 cas d'utilisation <<enseignant>>

Le schéma de la figure …. illustre le cas d'utilisation « Enseignant » et les fonctionnalités réalisables par l'administrateur dans le système. Le tableau suivant présente la description de ce cas d'utilisation.

|  |  |
| --- | --- |
| **Case** | **Description** |
| Authentification | L'utilisateur s'authentifie en tant qu'enseignant pour accéder au site Web |
| deposement des news | L'enseignant publie des actualités |
| deposement des notes | L'enseignant partage les notes des étudiants |

#### Diagramme de cas d’utilisation d’étudiant

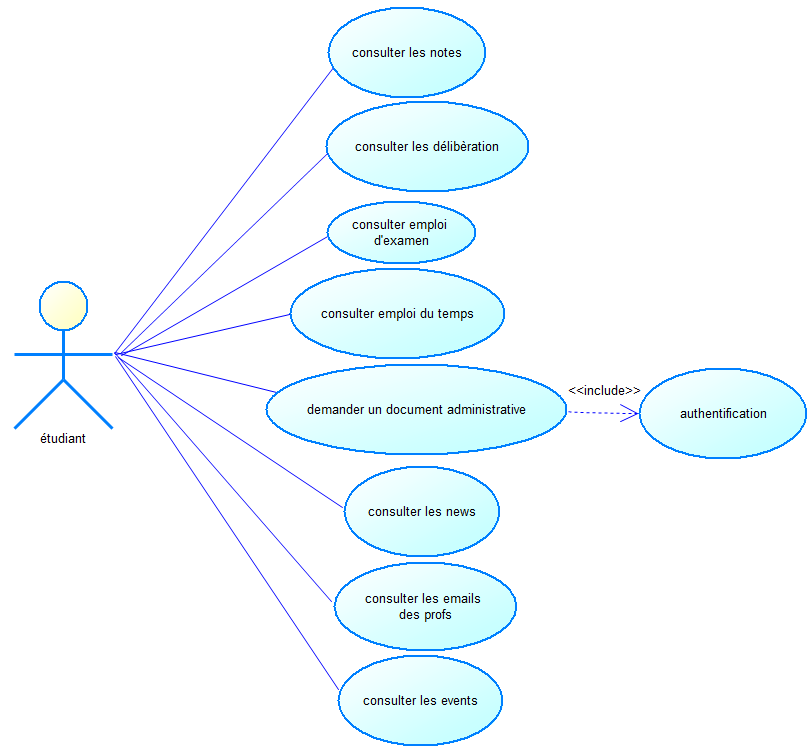


Figure ‎2.5‑5 cas d'utilisation <<student>>

Le schéma de la figure …. illustre le cas d'utilisation « étudiant » et les fonctionnalités réalisables par l'administrateur dans le système. Le tableau suivant présente la description de ce cas d'utilisation.

|  |  |
| --- | --- |
| **Case** | **Description** |
| Authentification | L'utilisateur s'authentifie en tant qu'étudiant pour accéder au site Web |
| consultation des notes | L'étudiant visualise ses notes académiques |
| consultation des plannings | L'étudiant vérifie son planning (EDT,Examens) |
| consultation des délibérations | L'étudiant visualise les délibérations. |
| consultation des actualités | L'étudiant lit les events et les news |
| consultation les emails | L'étudiant voit les emails de ses professeurs |
| Demande d’un document administrative | L'étudiant demende un document en-ligne |

## Diagramme de séquence

### Definition

Le diagramme de séquence est un type de diagramme UML (Unified Modeling Language) qui modélise **les interactions dynamiques entre objets** dans un système, en se focalisant sur **l'ordre chronologique des messages échangés**.Il indique les objets que l’acteur va manipuler et les opérations qui font passer d’un objet à l’autre. Il sert à illustrer un cas d’utilisation.

Dans cette section, nous nous intéressons à présenter, pour quelques cas d’utilisation, les diagrammes de séquence système correspondants

-Le tableau suivant représente les principaux concepts du diagramme de séquence.

|  |  |  |
| --- | --- | --- |
| Attribut | Symbole | Definition |
| boite d’activation |  | Représenté par un rectangle en haut d'une ligne de vie ; il peut s'agir d'un acteur externe ou d'un objet système. |
| Acteur / Objet / Classe |  | Représenté par un rectangle en haut d'une ligne de vie ; il peut s'agir d'un acteur externe ou d'un objet système. |
| Ligne de vie |  | Ligne verticale qui représente la durée de vie d’un objet/acteur pendant l’interaction. |
| alt operateur |  | Utilisé pour représenter **plusieurs chemins possibles** dans l’interaction, selon une **condition**. Chaque chemin (appelé "operand") a une **condition de garde** (guard condition). Ex : if (condition) … else … |
| ref operateur |  | Sert à **référencer un autre diagramme de séquence**. Très utile pour **réutiliser** des interactions complexes ou pour **modulariser**. |
| Message synchrone |  | Flèche pleine avec pointe fermée, indiquant un appel bloquant (appel de méthode). |
| Message Asynchrone |  | Flèche pleine avec pointe ouverte, indiquant un appel non bloquant. |
| Message de retour |  | Ligne pointillée avec flèche, indiquant une réponse ou un retour de méthode. |

Figure ‎2.6‑1les principaux concepts du diagramme de séquence

### Diagramme de séquence de cas d’utilisation <<authentification>>

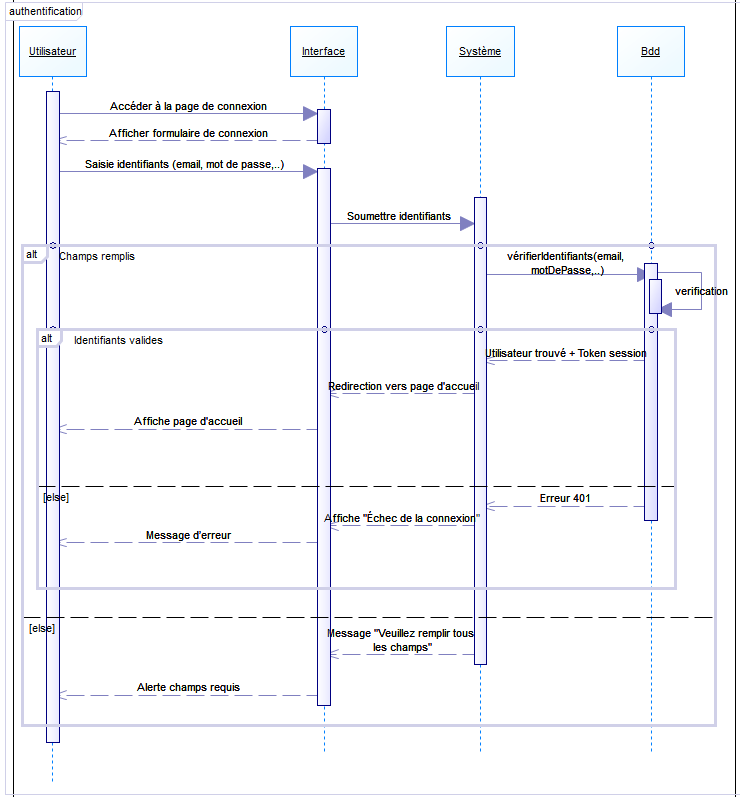
****

Figure ‎2.6‑2 Diagramme de séquence de cas d’utilisation <<authentification>>

### Diagramme de séquence de cas d’utilisation <<Déposé les notes>>

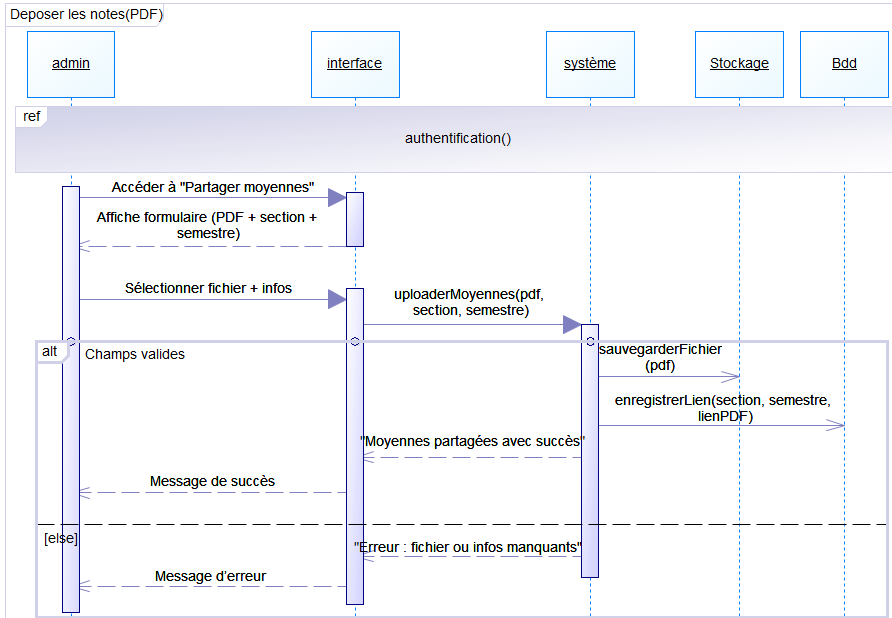
****

Figure ‎2.6‑3 Diagramme de séquence de cas d’utilisation <<Déposé les notes>>

### Diagramme de séquence de cas d’utilisation <<Demande de document>>

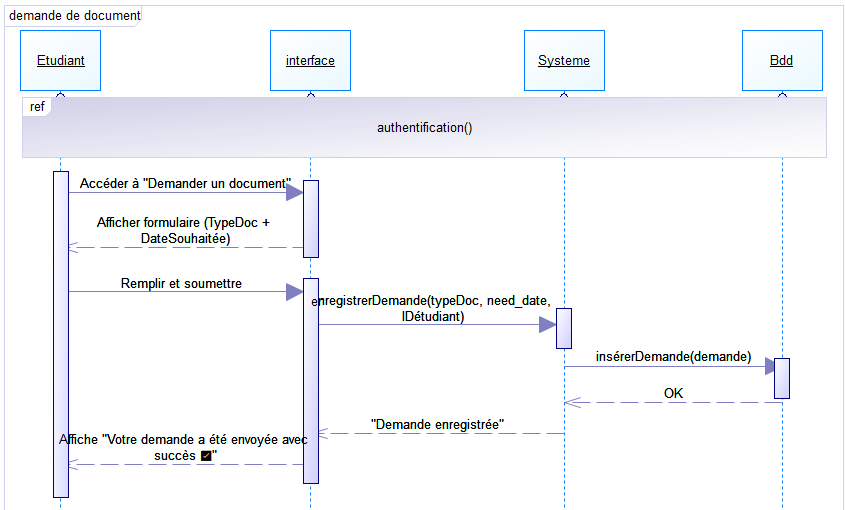
****

Figure ‎2.6‑4 Diagramme de séquence de cas d’utilisation <<Demande de document>>

### Diagramme de séquence de cas d’utilisation <<Gérer demande de document>>

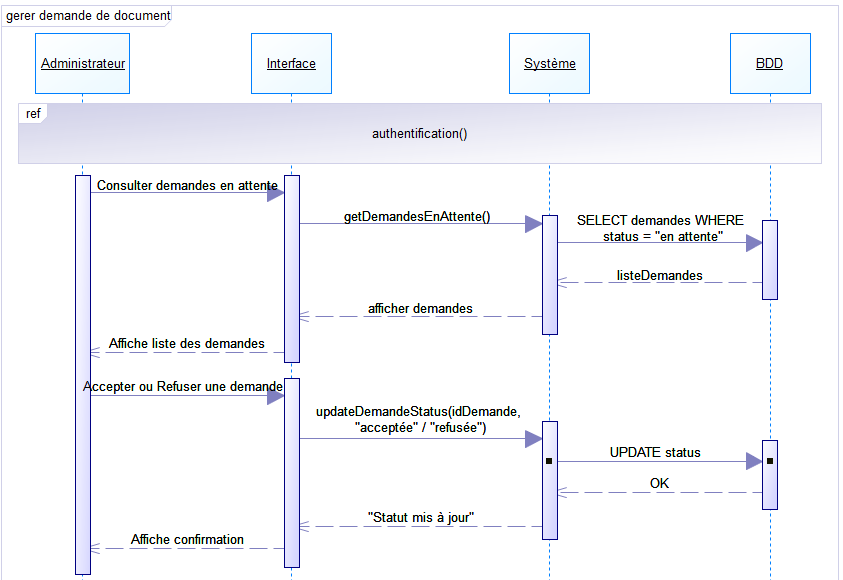
****

Figure ‎2.6‑5 Diagramme de séquence de cas d’utilisation <<Gérer demande de document>>

### Diagramme de séquence de cas d’utilisation <<Déposer cour>>

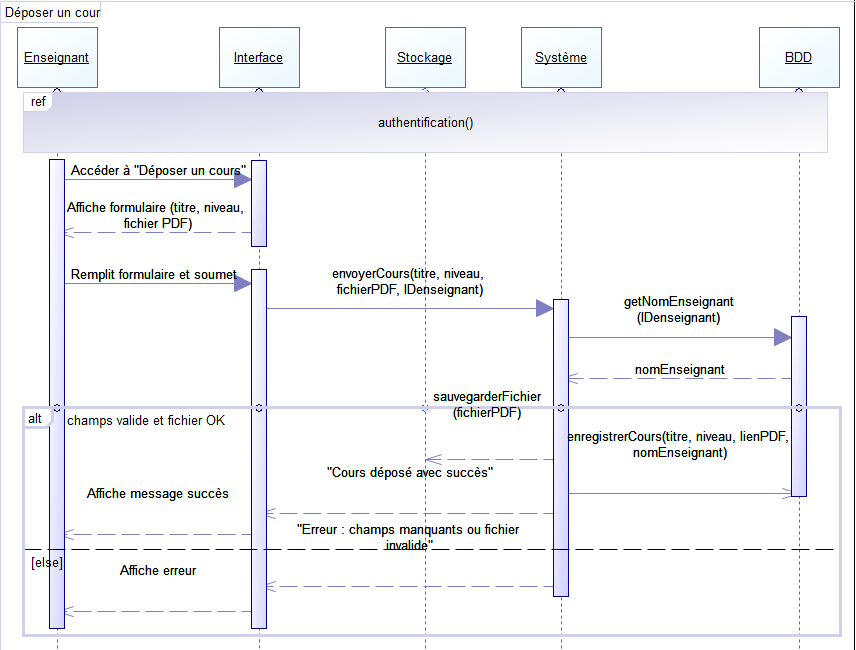


Figure ‎2.6‑6 Diagramme de séquence de cas d’utilisation <<Déposer cour>>

## dictionnaire des données

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributs** | **Type** | **Taille** | **Description** |
| id | **int** | **100** | **identifiant d’utilisateur** |
| First\_Name | **Varchar** | **30** | **prenom d’utilisateur** |
| Family\_Name | **Varchar** | **30** | **nom d’utilisateur** |
| password | **Varchar** | **30** | **mot de passe d’utilisateur** |
| role | **Varchar** | **10** | **role d’utilisateur** |
| student\_id | **int** | **100** | **identifiant de l’étudiant** |
| matricule | **int** | **100** | **matricule de l’étudiant** |
| student\_level | **Varchar** | **10** | **niveaux de l’étudiant** |
| student\_speciality | **Varchar** | **10** | **specialité de l’étudiant** |
| **teacher\_id** | **int** | **100** | **l’identifiant d’enseignant** |
| **teacher\_email** | **Varchar** | **100** | **l’émail d’enseignant** |
| **admin\_id** | **int** | **100** | **l’identifiant d’admin** |
| **admin\_email** | **Varchar** | **100** | **l’émail d’admin** |
| **lecture\_id** | **int** | **100** | **l’identifiant de cour** |
| **lecture\_title** | **Varchar** | **50** | **le titre de cour** |
| **lecture\_level** | **VarChar** | **10** | **le niveau de cour** |
| **upload\_date** | **Date** | **20** | **la date de partage** |
| **lecture\_filepath** | **VarChar** | **100** | **l’emplacement de fichier de cour** |
| **lecture\_publisher** | **VarChar** | **100** | **le partageur de cour** |
| **grade\_id** | **int** | **100** | **l’identifiant de note** |
| **grade\_filepath** | **VarChar** | **100** | **l’emplacement de fichier de note** |
| **grade\_level** | **VarChar** | **10** | **le niveau de note** |
| **grade\_semester** | **VarChar** | **10** | **le semestre de note** |
| **schedule\_id** | **int** | **100** | **l’identifiant de EDT** |
| **schedule\_level** | **Varchar** | **10** | **le niveau de EDT** |
| **schedule\_semester** | **Varchar** | **10** | **semester de EDT** |
| **creationDate** | **Varchar** | **20** | **la date de creation** |
| **schedule\_filepath** | **Varchar** | **100** | **l’emplacement de fichier de EDT** |
| **schedule\_type** | **Varchar** | **10** | **le type de EDT** |
| **news\_id** | **int** | **100** | **l’identifiant de news** |
| **target\_level** | **Varchar** | **50** | **l’audiance specifié pour recevoir le news** |
| **news\_content** | **Varchar** | **1000** | **le contenu de news** |
| **news\_date\_posted** | **Varchar** | **20** | **la date de partage de news** |
| **news\_publisher** | **Varchar** | **50** | **la partageur de news** |
| **event\_id** | **int** | **100** | **l’identifiant d’event** |
| **event\_title** | **Varchar** | **50** | **le titre d’event** |
| **event\_content** | **Varchar** | **1000** | **le contenu d’event** |
| **image\_path** | **/** |  | **l’emplacement d’image d’event** |
| **event\_date\_posted** | **Varchar** | **20** | **la date de partage d’event** |
| **request\_id** | **int** | **100** | **l’identifiant de demande** |
| **need\_date** | **Varchar** | **20** | **la date de besoin** |
| **document\_request\_type** | **Varchar** | **50** | **le type de document demandé** |
| **status** | **Varchar** | **50** | **le statu de document** |
| **requestCreationDate** | **Varchar** | **20** | **la date de creation de demande** |

Figure ‎2.10‑1 Description des tables de la BDD (MYSQL)

## Règles de transformation du diagramme de classes en modèle relationnel

Les règles utilisées pour obtenir le diagramme de classes de notre application sont les suivantes:

* Règle 1 : Chaque classe devient une table, les attributs de la classe deviennent des attributs et l'identifiant devient une clé primaire pour la table.
* Règle 2 : Chaque association 1-1 est prise en compte en incluant la clé primaire de l’une des relations comme clé étrangère dans l’autre relation.
* Règle 3 : Chaque association 1-N est prise en compte en incluant la clé comme clé étrangère dans la relation dont la multiplicité maximale est égale à la clé primaire de l’autre relation.
* Règle 4 : Chaque association M-N entraîne la création d’une relation dont la clé primaire est composée de clés étrangères référençant les relations correspondant aux entités liées par l’association. Toutes les propriétés de l’association deviennent des attributs de la relation.
* Règle 5 : Transformation de la relation d’héritage

Distinction : les classes seront des tables, les attributs des colonnes et les identifiants

les clés primaires, la clé primaire de la classe elle-même migrera vers la classe enfant comme

une clé étrangère.

Push-down : On migre tous les attributs de la classe parent vers la classe enfant, par conséquent on ne conserve qu'une seule classe qui est la classe enfant.

Push-up : On migre tous les attributs des classes enfants vers la classe parent.

## Diagramme de classe

Le diagramme de classes est un type de diagramme structurel en UML (Unified Modeling Language) qui représente la structure statique d’un système.Il décrit les classes, leurs attributs, leurs méthodes (opérations), ainsi que les relations entre elles (comme l’héritage, l'association, la composition, etc.).Le diagramme de classe permet d’exprimer comment les objets et les classes vont être définies, ainsi que les relations qui existent entre les différentes classes.

Le diagramme de classe de notre système est donné dans la figure :

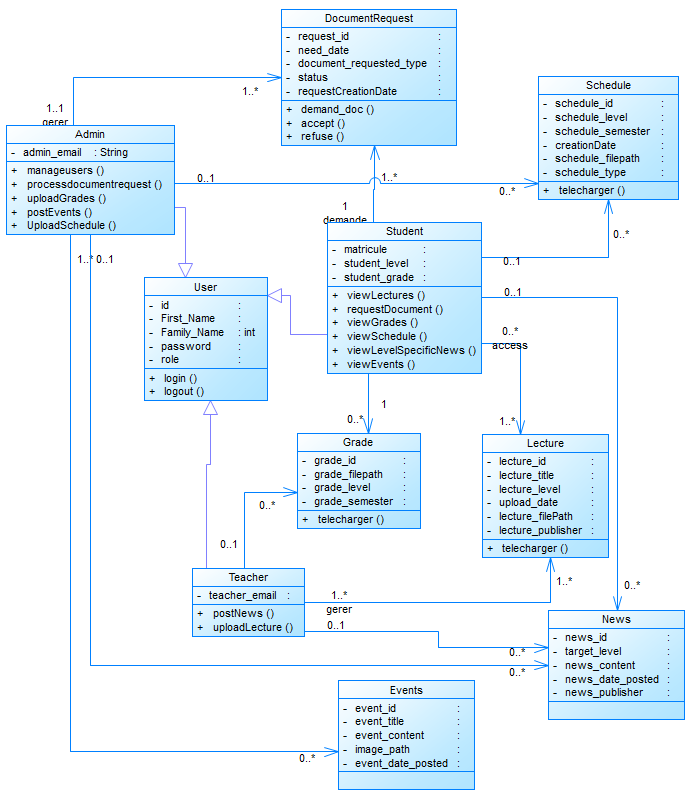
****

Figure ‎2.7‑1 Diagramme de classe

## Modèle relationnel

En appliquant les règles de passage du modèle objet au modèle relationnel sur notre diagramme de classe, nous obtenons les relations suivantes :

Admin (**admin\_id**, email,name,password,role).

Student (**student\_id**, student\_level, student\_grade,name,password,role).

Teacher (**teacher\_id**, email,name,password,role).

Lecture (**lecture\_id**, **#teacher\_id,** lecture\_title, lecture\_level,upload\_date,lecture\_filepath, lecture\_publisher).

Schedule (**schedule\_id**, schedule\_level, semester, creation\_date, schedule\_filepath, schedule\_type).

Grade (**grade\_id**, grade\_filepath, grade\_level, grade\_semester).

News (**news\_id**, **#teacher\_id,** target\_level, content, date\_poster, news\_publisher).

Events (**event\_id**, title, content, image\_path, date\_posted).

DocumentRequest (**request\_id**, **#student\_id,** need\_date, document\_requested\_type, studentId, status, requestCreationDate).

## Conclusion

Dans ce chapitre, nous avons modélisé notre plateforme à l’aide de diagrammes UML pour définir son fonctionnement et sa structure. Cette étape a guidé la conception de notre base de données et posé les bases de l’implémentation. Le prochain chapitre sera dédié au développement de l’application, aux outils utilisés et aux interfaces réalisées.

# Implémentation

## Intoduction

Après avoir défini et modélisé notre application, ce chapitre est consacré à sa **concrétisation**. Nous y présentons les **technologies** et **méthodes** utilisées pour le développement de la plateforme, ainsi qu’un aperçu des principales **interfaces** et **fonctionnalités** mises en place.

## Outils, langages et technologies de développement

### Les langages de balisage, de style et de programmation

#### HTML 5

**HTML** (HyperText Markup Language) est un langage de balisage utilisé pour structurer le contenu des pages web. Il permet d’organiser les éléments tels que les titres, les paragraphes, les liens, les images et autres composants visibles sur une page web. C’est la base de toute page web sur Internet.

#### CSS 3

**CSS** (Cascading Style Sheets) est un langage de style utilisé pour décrire la présentation et la mise en forme des pages web écrites en HTML. Il permet de contrôler l’apparence des éléments HTML, comme les couleurs, les polices, les espacements, les dispositions, et bien plus. Grâce à CSS, les concepteurs peuvent créer des sites web esthétiquement plaisants et adaptés à différents types d’appareils.

#### JAVASCRIPT

**JavaScript** est un langage de programmation utilisé principalement côté client pour rendre les pages web interactives. Il permet de manipuler le contenu, de gérer les événements utilisateurs (clics, saisie) et d'effectuer des mises à jour sans recharger la page. C'est un outil clé pour le développement front-end des sites web modernes.

#### PHP

**PHP (Hypertext Preprocessor)** est un langage de script côté serveur utilisé pour créer des pages web dynamiques. Il permet aux développeurs d'intégrer du code au sein du HTML pour générer du contenu en fonction des entrées utilisateur ou des requêtes de base de données. PHP est largement utilisé pour des tâches telles que la gestion des formulaires, l'interaction avec des bases de données et la gestion des sessions. C'est un outil puissant pour créer des sites web interactifs et basés sur des données.

#### SQL

**SQL (Structured Query Language)** est un langage utilisé pour gérer et manipuler les bases de données relationnelles. Il permet de créer, modifier, et interroger des bases de données afin d'extraire ou de manipuler des données. SQL est essentiel pour toute application web nécessitant une gestion dynamique de données, comme la gestion des utilisateurs ou l'organisation des fichiers dans une plateforme.

### Environnements de développement et outils utilisés

#### VSCODE

**Visual Studio Code (VSCode)** est un éditeur de code source léger et puissant, développé par Microsoft. Il prend en charge de nombreux langages et offre des fonctionnalités comme l’autocomplétion, le débogage, et l’intégration de Git, ce qui en fait un outil populaire pour le développement web.

#### XAMPP

**XAMPP** est un environnement de développement local qui permet d'exécuter des serveurs web sur votre ordinateur. Il inclut Apache, MySQL, PHP, et Perl, facilitant ainsi le développement et les tests d’applications web avant leur déploiement sur un serveur distant.

#### PhpMyAdmin

**phpMyAdmin** est une application web permettant de gérer facilement des bases de données MySQL ou MariaDB via une interface graphique. Elle offre des fonctionnalités telles que la création, la modification et la suppression de bases de données, ainsi que l'exécution de requêtes SQL, le tout de manière intuitive.

#### Bootstrap

**Bootstrap** est un framework CSS open-source conçu pour faciliter et accélérer le développement de sites web responsives et modernes. Il fournit un ensemble de classes prédéfinies, de composants interactifs, et de grilles flexibles permettant d’assurer une mise en page cohérente et adaptée à tous les types d’écrans sans écrire beaucoup de code personnalisé.

#### PowerDesigner

**PowerDesigner** est un outil de modélisation développé par SAP, utilisé principalement pour la conception de bases de données et la modélisation UML. Il permet de créer des diagrammes comme les modèles conceptuels, logiques et physiques de données, ainsi que des diagrammes de classes et de séquences.

## Interfaces de l’application

